**Software Project Management Plan**

**GTA Application Portal**

12/08/2023

**Team Members**

Seth Emery

Blake Lord

Brandon Piotrowski

Ryan Borland

Document Control

**Change History**

| **Revision** | **Change Date** | **Description of changes** |
| --- | --- | --- |
| V1.0 | 11/29/2023 | Initial release |
| V2.0 | 12/08/2023 | Final Revisions |

[Note change history doesn’t have to be stored with the document. Most version control tools like SVN keep track of change history automatically. The comments entered when checking in documents become the change history for the document. However, even if your version control tool supports change history, you make want to track a more detailed version here.]

**Document Storage**

This document is stored in the project’s SVN repository at: http://company.com/svn/project-name/docs/spmp.doc.

**Document Owner**

Brandon Piotrowski is responsible for developing and maintaining this document.**Table of Contents**

[**1**](#_3znysh7) **OVERVIEW 4**

[**1.1**](#_49x2ik5) **Purpose and Scope 4**

[**1.2**](#_2p2csry) **Goals and Objectives 4**

[**1.3**](#_3dy6vkm) **Project Deliverables 5**

[**1.4**](#_1t3h5sf) **Assumptions and Constraints 5**

[**1.5**](#_4d34og8) **Schedule and Budget Summary 6**

[**1.6**](#_2s8eyo1) **Success Criteria 6**

[**1.7**](#_17dp8vu) **Definitions 6**

[**1.8**](#_3rdcrjn) **Evolution of the Project Plan 6**

[**2**](#_26in1rg) **STARTUP PLAN 6**

[**2.1**](#_lnxbz9) **Team Organization 6**

[**2.2**](#_35nkun2) **Project Communications 7**

[**2.3**](#_1ksv4uv) **Technical Process 7**

[**2.4**](#_44sinio) **Tools 7**

[**3**](#_2jxsxqh) **WORK PLAN 7**

[**3.1**](#_z337ya) **Activities and Tasks 7**

[**3.2**](#_3j2qqm3) **Release Plan 8**

[**3.3**](#_1y810tw) **Iteration Plans 8**

[**3.4**](#_4i7ojhp) **Budget 8**

[**4**](#_2xcytpi) **CONTROL PLAN 8**

[**4.1**](#_1ci93xb) **Monitoring and Control 8**

[**4.2**](#_3whwml4) **Metrics Collection 8**

[**5**](#_2bn6wsx) **SUPPORTING PROCESS PLANS 8**

[**5.1**](#_qsh70q) **Risk Management Plan 8**

[**5.2**](#_3as4poj) **Configuration Management Plan 8**

[**5.3**](#_1pxezwc) **Verification and Validation Plan 9**

# **Overview**

## *Purpose and Scope*

The purpose of the GTA Application Portal is to provide UMKC students and administrators a streamlined way to manage applications in order to save time and increase the number of applicants applying to these positions. Having an easy-to-use and simple job application process will improve the number of completed applications being submitted.

The GTA Application Portal will include a user account system along with methods to submit and edit applications. Administrative users will need to be able to create new job listings, edit existing job listings, manage student user submitted applications, and view all other user accounts. Documentation will be provided along with this project which will include a user guide that explains not only how to use the website, but also how to set up and install the program.

## *Goals and Objectives*

Project Goals:

1. Provide an easy-to-use website for submitting and storing applications
2. Follow similar design patterns of the UMKC website

Project Objectives:

1. Create a database that can store user accounts, applications, and job listing
2. Create a website that can access the data from the database
3. Create an interface for users to be able to register accounts that will be stored in the database
4. Create an interface for users to be able to log in to their account
5. Create an interface for users to be able to manage their account information (name, email, password, classes taken, etc.)
6. Create an interface for student users to be able to view and apply to job listings
7. Create an interface for administrative users to be able to post new job listings, manage existing job listings, manage student submitted applications, and view all users

## *Project Deliverables*

The following items will be delivered on or before 12/08/2023:

1. All source code for the project (Github)
2. Requirements Document
3. Project Plan Document
4. Architecture Document
5. Testing Documents:
   1. Test Plan
   2. Test Cases
   3. Test Report
6. Team Effort Estimation Document
7. User Guide Document (Including installation guide)
8. PowerPoint Presentation Document outlining how the project was created

## *Assumptions and Constraints*

Assumptions:

1. The user has access to the internet and a web browser
2. If running the application, the user has access to a text editor (preferably Visual Studio Code) and can install the required items listed in the repository

Constraints:

1. The product must run in a web browser
2. The product must be complete by 12/08/2023
3. The database must be SQLite
4. The frameworks being used are Flask and Bootstrap

## *Schedule* *Summary*

| **Start Date** | **End Date** | **Deliverable/Milestone** |
| --- | --- | --- |
| 08/25/2023 | 09/01/2023 | 1. Research project tech stack and tools |
| 09/01/2023 | 09/15/2023 | 1. SQLite database set up |
| 09/15/2023 | 09/22/2023 | 1. Basic website set up and paired with database |
| 09/22/2023 | 11/15/2023 | 1. User accounts set up (Register, login, account page) |
| 09/22/2023 | 11/15/2023 | 1. Job listings page and functionalities |
| 09/22/2023 | 11/15/2023 | 1. Application submission |
| 09/22/2023 | 12/08/2023 | 1. Administration page and functionalities |
| 08/25/2023 | 12/08/2023 | 1. Documentation Complete |
| 08/25/2023 | 12/08/2023 | 1. Project Finish |

## *Success Criteria*

* All high-priority use cases in the requirements document are delivered before 12/08/2023
* 95% of users are able to use the website with minimal or no issues
* All project deliverables are completed before 12/08/2023

## *Definitions*

**GTA Application Portal -** The website that is being described in this document

**User -** A person who interacts with the GTA Application Portal website

**Administrator** - A user who has authoritative access to the data stored in the database of the GTA Application Portal website. This user also has deeper access to the underlying systems of the website

**Student** - A user who is currently studying or pursuing an education at UMKC, and is logged in with minimal access to the underlying systems of the GTA Application Portal website

**Use Case** - describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

# **Startup Plan**

## *Team Organization*

| **Title** | **Members** | **Description** |
| --- | --- | --- |
| Project Manager | Seth Emery | The project manager is responsible for communication with the client (Professor Shah), organizing team meetings, and maintaining the schedule |
| Programmers | Seth Emery, Blake Lord, Brandon Piotrowski | The programmers are responsible for coding, unit testing, and designing all parts of the website. They are expected to communicate what they are working on next or currently during team meetings. |
| Researchers | Seth Emery, Blake Lord, Brandon Piotrowski,  Ryan Borland | The researchers are required to gather information about what is required of the project, the tech stack, and maintain documentation. They are expected to communicate any new information they have acquired during team meetings |

## *Project Communications*

The primary form of communication will be over Discord. Any resources acquired or documentation can be shared through this channel. Communication about the project is also expected during or after class times.

## *Technical Process*

Due to time constraints and team member obligations outside of the classroom like work, the team has decided to have a weekly team meeting after Friday class to discuss what they have accomplished, and are going to work on next, similar to a SCRUM meeting. Each team member will be given a number of tasks that need to be completed for the current project phase each week. To ensure no issues arise in the version control, members are expected to message the team over Discord when they perform a push on the github.

| **Phase** | **Entry Criteria** | **Exit Criteria** | **Milestone Accomplished (See Section 1.5)** |
| --- | --- | --- | --- |
| Research | N/A | Tech stack is decided on, version control is decided on, basic requirements have been compiled. | 1 |
| Database Initialization | Research Complete | Database tables are fully set up and ready to use. | 2 |
| Prototype Website | Database Initialization Complete | A temporary website is working and has access to the database. | 3 |
| User Account Registration | Prototype Website | Users can create a student account, and all of the corresponding information is stored in the database. | 4.1 |
| User Login Page | Prototype Website | Users can login to their account that is already registered and account login validation checking. | 4.2 |
| User Account Page | Prototype Website | Logged in users can view their account page and modify their information. | 4.3 |
| Job Listing Page | Prototype Website | Created jobs are displayed on the jobs page with its information and an apply button. Apply button takes them to the application page if logged in, otherwise redirects to the login page. | 5 |
| Application Page | Job Listing Page | Application page appears with the corresponding job that was clicked, user info auto fills but is able to be edited, file submission is provided. Redirects to the my applications page on submission. | 6.1 |
| My Applications Page | Job Listing Page, Application Page | My application page appears when the My Applications tab is clicked. The current users submitted applications are viewable with an edit button or a message displaying uneditable. | 6.2 |
| Administrator Page | Job Listing Page, User Accounts | Administration page appears when the Administration tab is clicked. A create job button is present, list of all current job listings along with an edit button is present, list of all current student submitted applications along with an edit/view, open/close, accept, and reject button is present, a list of all user accounts is present. | 7.1 |
| Create New Job Listing | Job Listing Page | A new page appears when the create job button is pressed on the administration page. inputs to select the class, job position, and certification requirements are present. After submission the job is listed on the jobs listings page. | 7.2 |
| Edit Existing Job Listing | Job Listing Page | A new page appears when the edit button is clicked on a job within the administration page. The administrator can change the class for the job, the job position, and the certification requirements. After submission the job listing is updated on the jobs listings page. | 7.3 |
| Manage Student Applications | Job Listing Page, Applications | When the edit/view button is clicked a page opens with uneditable input boxes displaying everything the student submitted. When the open/close button is clicked the student's application is either opened for editing or closed for no editing. When the accept button is clicked the application, job listing, and all other applications related to that job listing are removed from the database. When the reject button is clicked only that application is removed from the database. | 7.4 |
| Documents | Research Complete | Requirements documentation, project plan documentation, architecture documentation, test plan documents, test cases documents, test reports documents, team effort estimation documentation, user guide documentation, and project powerpoint slides are complete | 8 |

## *Tools*

**Programming Languages** - Python, HTML, CSS, JavaScript

**Frameworks** - Flask, Bootstrap

**Databases** - SQLite

**Version Control** - Github repository

**IDE/Text Editor** - Visual Studio Code

**Automated Testing** - PyTest, Playwright

**Task/Bug Tracking** - Trello

# **Work Plan**

## *Activities and Tasks*

See Team Effort Estimation Document.

## *Release Plan*

| **Date** | **Deliverable/Milestone** |
| --- | --- |
| 09/15/2023 | SQLite database set up |
| 09/22/2023 | Basic website set up and paired with database |
| 11/15/2023 | User accounts set up (Register, login, account page) |
| 11/15/2023 | Job listings page and functionalities |
| 11/15/2023 | Application submission |
| 12/08/2023 | Administration page and functionalities |
| 12/08/2023 | Documentation Complete |
| 12/08/2023 | Project Finish |

## *Iteration Plans*

See Team Effort Estimation document.

# **Control Plan**

## *Monitoring and Control*

Weekly - Team Meeting. Group will discuss what has been accomplished over the previous week, what to work on next, and any problems discovered.

09/15/2023 - Database setup review. Group will review the database to ensure understanding amongst group members.

09/22/2023 - Website prototype review. Group will review code patterns being utilized and database accessibility.

11/15/2023 - User accounts review. Group will review functionalities of user accounts and ensure the database is storing information properly.

11/15/2023 - Job listing review. Group will review functionalities of job listings and ensure the database is storing information properly

11/15/2023 - Applications review. Group will review functionalities of applications and ensure the database is storing information properly

12/08/2023 - Administration page review. Group will review all functionalities of the administration page and ensure the database is storing information properly

12/08/2023 - Documentation review. Group will review all documentation to ensure that it is complete and accurate.

## *Project Measurements*

| **Phase** | **Measurement** | **Source** |
| --- | --- | --- |
| Release Planning | Record effort estimates for product features. | Project Manager, Programmers |
| Iteration Planning | Record effort estimates for scheduled tasks.  Update effort estimates for product features.  Update estimated dates in release plan. | Project Manager, Programmers |
| Iteration Closeout | Record actual effort for scheduled tasks.  Record actual effort for product features. | Project Manager, Programmers |
| System Test | Record the rate at which errors are found.  Update the testing documents. | Project Manager, Programmers |

# **Supporting Process Plans**

## *Risk Management Plan*

Risks:

1. Unauthorized administration page access. Access to pages will no longer be allowed through web page input (manually typing the page into the search bar). Some consequences that could occur if this risk is not mitigated are jobs being removed and student applications being accepted or rejected without review.
2. Unauthorized user submitting a job application. An attempt to apply to a job without being logged in will automatically redirect the user to the login page. If this risk is not handled it could result in a DDoS attack if a user automates applications and submits an abnormally large number of applications.
3. Administrator accounts can only be created in the backend. This ensures that no one can register as an administrator and gain access to the administration page.

## *Configuration Management Plan*

1. All files will be stored in the projects Github repository
2. Installation requirements will contain its own file to allow for easy updating and installation for group members.
3. All members of the group must be notified whenever a git push is performed
4. A phase will only be considered completed when all group members have reviewed completed tasks.

## *Verification and Validation Plan*

The verification and validation of the product will occur throughout the project's lifecycle. After a new feature is implemented group members will discuss what needs to be changed, if anything, and will each perform their own manual testing on it to ensure the feature functions as intended.

## *Product Acceptance Plan*

The product will be accepted if all requirements are fulfilled by 12/08/2023 and approximately 95% of users can use the product with minimal or no issues.